Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление 09.03.01 – «Информатика и вычислительная техника»

Дисциплина: «Защита информации»

Профиль: «Разработка программно-информационных систем»

Семестр 5

ОТЧЕТ

по лабораторной работе №3

Тема: «Несимметричные алгоритмы шифрования»

Выполнил: студент группы РИС 20-2б

Уржумов В.И. \_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ИТАС

Шереметьев В. Г. \_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_

Пермь, 2022

**ЦЕЛЬ РАБОТЫ**

Получить практические навыки по использованию несимметричных алгоритмов шифрования, на примере использования алгоритма Диффи-Хеллмана.

**ЗАДАНИЕ**

Выполнить шифрование текстового файла методом Диффи-Хеллмана, используя в качестве x и y простые числа с разрядностью не меньшей двенадцати, выполнив условие случайности x и y для каждого нового шифрования.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**Алгоритм Диффи-Хеллмана:**

Алгоритм назван по фамилиям его создателей Диффи (Diffie) и Хеллмана (Hellman).

Метод помогает обмениваться секретным ключом для симметричных криптосистем, но использует метод, очень похожий на асимметричный алгоритм RSA. Это не симметричный алгоритм, так как для шифрования и дешифрования используются различные ключи. Так же это не схема с открытым ключом, потому что ключи легко получаются один из другого, и ключ шифрования и ключ дешифрования должны храниться в секрете.

Определим круг его возможностей. Предположим, что двум абонентам необходимо провести конфиденциальную переписку, а в их распоряжении нет первоначально оговоренного секретного ключа. Однако, между ними существует канал, защищенный от модификации, то есть данные, передаваемые по нему, могут быть прослушаны, но не изменены (такие условия имеют место довольно часто). В этом случае две стороны могут создать одинаковый секретный ключ, ни разу не передав его по сети, по следующему алгоритму.

Предположим, что обоим абонентам известны некоторые два числа v и q. Они, впрочем, известны и всем остальным заинтересованным лицам. Например, они могут быть просто фиксированно «зашиты» в программное обеспечение. Далее один из партнеров P1 генерирует случайное или псевдослучайное простое число x и посылает другому участнику будущих обменов P2 значение A = qx mod n

По получении А партнер P2 генерирует случайное или псевдослучайное простое число у и посылает P2 вычисленное значение B = qy mod n

Партнер P1, получив В, вычисляет Kx = Bx mod n, а партнер P2 вычисляет Ky = Ay mod n. Алгоритм гарантирует, что числа Ky и Kx равны и могут быть использованы в качестве секретного ключа для шифрования. Ведь даже перехватив числа А и В, трудно вычислить Kx или Ky.

Например, по вычисленным Kx =Ky=K абоненты могут зашифровать сообщение M=123 по следующему алгоритму: к каждому символу сообщения M добавить K => сообщение С=234, при K=1. Соответственно алгоритмом расшифрования будет разность ключа K из каждого символа сообщения C.

Пример:

Пусть

n=3; q=5;

x=5; y=7;

тогда A = q^x mod n = 1, а B = q^y mod n = 2, то вычислив Kx = B^x mod n и Ky = A^y mod n получим Kx= Ky=1. Зашифруем приведенное выше сообщение M=123 по приведенному выше алгоритму => сообщение С=234, расшифровав сообщение C по обратному алгоритму получим сообщение M=123.

Необходимо еще раз отметить, что алгоритм Диффи-Хеллмана работает только на линиях связи, надежно защищенных от модификации. Если бы он был применим на любых открытых каналах, то давно снял бы проблему распространения ключей и, возможно, заменил собой всю асимметричную криптографию. Однако, в тех случаях, когда в канале возможна модификация данных, появляется очевидная возможность вклинивания в процесс генерации ключей «злоумышленника-посредника» по той же самой схеме, что и для асимметричной криптографии.

**ХОД РАБОТЫ**

На рисунке 1 представлена главная форма программы. Пример работы программы представлен на рисунке 2.

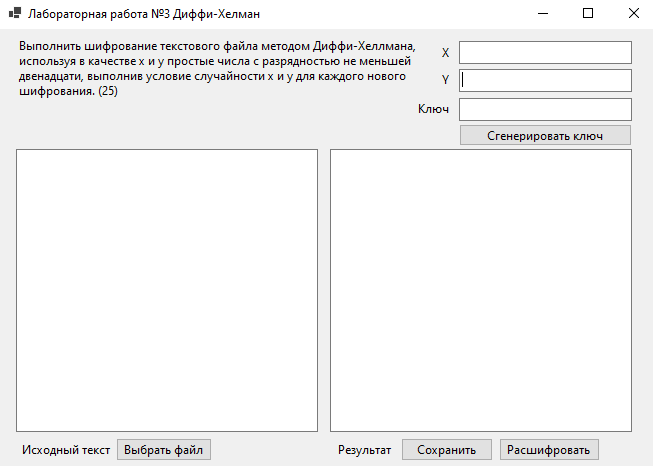


Рисунок 1- Главная форма

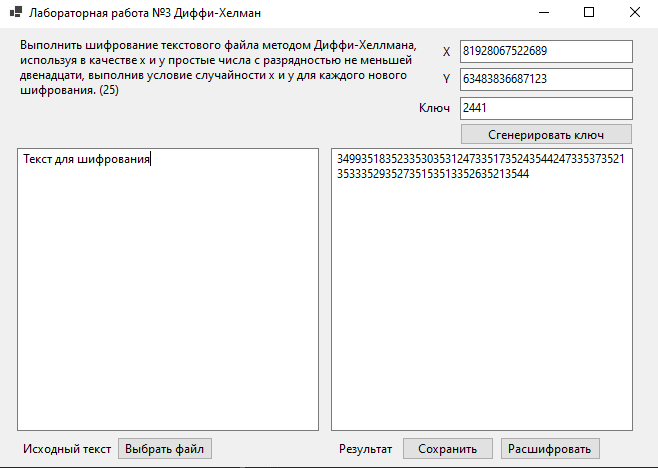


Рисунок 2-Пример работы программы

**ПРИЛОЖЕНИЕ А**

**Листинг класса Labs3**

public partial class labs3 : Form

{

Generator g = new Generator();

BigInteger x=0;

BigInteger y=0;

BigInteger gk=0;

BigInteger pk=0;

ulong secretKey = 0;

public labs3()

{

InitializeComponent();

}

public BigInteger GetPRoot(BigInteger p)

{

for (BigInteger i = 0; i < p; i++)

if (IsPRoot(p, i))

return i;

return 0;

}

public bool IsPRoot(BigInteger p, BigInteger a)

{

if (a == 0 || a == 1)

return false;

BigInteger last = 1;

HashSet<BigInteger> set = new HashSet<BigInteger>();

for (BigInteger i = 0; i < p - 1; i++)

{

last = (last \* a) % p;

if (set.Contains(last)) // Если повтор

return false;

set.Add(last);

}

return true;

}

private void button3\_Click(object sender, EventArgs e)

{

Random rand = new Random();

x= g.Generate();

y= g.Generate();

gk = rand.Next(1, 10000);

pk = rand.Next(1, 10000);

xPrime.Text = x.ToString();

yPrime.Text = y.ToString();

secretKey = (ulong)g.fastPow(x \* y, gk, pk);

textBox6.Text = secretKey.ToString();

if (textBox1.Text != "")

textBox2.Text = Encryped();

}

private void labs3\_FormClosed(object sender, FormClosedEventArgs e)

{

MainForm form1 = new MainForm();

form1.Show();

}

private void OpenFile()

{

OpenFileDialog openFileDialog1 = new OpenFileDialog();

openFileDialog1.Filter = "txt file (\*.txt)|\*.txt";

openFileDialog1.RestoreDirectory = true;

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

StreamReader sr = new StreamReader(openFileDialog1.FileName);

textBox1.Text = sr.ReadToEnd();

sr.Close();

}

}

private void SaveFile()

{

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

saveFileDialog1.Filter = "Text file(\*.txt)|\*.txt";

saveFileDialog1.RestoreDirectory = true;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

StreamWriter sw = new StreamWriter(saveFileDialog1.FileName);

sw.WriteLine(textBox2.Text);

sw.Close();

}

}

private void button1\_Click(object sender, EventArgs e)

{

OpenFile();

}

private void button2\_Click(object sender, EventArgs e)

{

SaveFile();

}

public string Encryped()

{

string encrypedMessage = "";

char[] message = textBox1.Text.ToCharArray();

foreach (char c in message)

{

encrypedMessage += (c + secretKey);

}

return encrypedMessage;

}

public string Decryped()

{

string secret = secretKey.ToString();

string temp="";

string encrypedMessage = "";

char[] ci = new char[5];

char[] message = textBox2.Text.ToCharArray();

foreach (char c in message)

{

temp += c.ToString();

if (temp.Length == secret.Length)

{

char ctmp = (char)(Convert.ToInt64(temp) -(long)secretKey);

encrypedMessage += ctmp.ToString();

temp = "";

}

}

return encrypedMessage;

}

private void button4\_Click(object sender, EventArgs e)

{

textBox2.Text = Decryped();

button4.Enabled = false;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

if (secretKey !=0)

textBox2.Text = Encryped();

button4.Enabled = true;

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

if (textBox6.Text == "")

secretKey = 0;

else

secretKey=Convert.ToUInt64(textBox6.Text);

textBox2.Text = Encryped();

}

}

**Листинг класса Generator**

public ulong Generate()

{

Random rand = new Random();

byte[] buffer = new byte[8];

rand.NextBytes(buffer);

ulong result = (ulong)Math.Abs(BitConverter.ToInt64(buffer, 0) / 100000);

while (true)

{

if (is\_prime(result) == false)

result++;

else

break;

}

return result;

}

public static bool is\_prime(ulong n)

{

if (n == 2)

return true;

if (n == 1)

return false;

for (uint i = 2; i <= Math.Sqrt(n); i++)

{

if (n % i == 0)

return false;

}

return true;

}

public BigInteger fastPow(BigInteger n, BigInteger pow, BigInteger mod)

{

BigInteger v = 1;

while (pow != 0)

{

if ((pow & 1) != 0)

v \*= n;

n \*= n;

pow >>= 1;

}

return v % mod;

}

}